BIRZEIT UNIVERSITY

First semester 2017/2018

SWEN7302, SECURE SOFTWARE DEVELOPMENT

Homework Assignment # 1

Due to Wednesday, December 06, 2017

Your answer should be brief and direct.

1. Why is waiting to think about security until after the software is built a bad idea?
2. What is an **abuse case**?
3. What is a reason for making an explicit threat model when designing a system?
4. Suppose you design software for a bank and the bank's customers may remotely log into its site using commodity PCs. These PCs might have malware on them, which could log keystrokes or read files stored on the machine. Which threat model makes the most sense for you to consider, when designing the bank's site?
5. What is a good defense against powers that are particular to a *snooping user*?
6. A denial of service attack violates what security *policy*/*goal*?
7. When talking about computer security, what do we mean by the term, **principal**?
8. Passwords, biometrics, and user-owned SMS-receiving mobile phones are useful for what security mechanism?
9. We identified three categories of secure design principles: *prevention*, *mitigation*, and *recovery*. Running each browser tab in a separate OS process (as done by the Chrome browser) is an example design illustrating which category?
10. Suppose you are implementing a graphical user interface for using a library implementing the RSA cryptosystem, and you want to give users a way to generate new keys. How do you do that taking security designs into account?
11. Suppose you are implementing an extensible data management system. You want to accommodate plug-ins that can implement storage rules and query processing functionality for different data formats (e.g., relational data, object data, XML data, etc.). How do you do that considering the security designs?
12. Promoting privacy is a goal that follows from which category of secure design principle?
13. Encrypting a password database is an example of what category of design principle?

14. Which of the following vulnerabilities can VSFTPD's secure string library help protect against?

15. VSFTPD forks a new process to handle each client connection. It could have, instead, spawned a thread within the main process to handle each connection, as is done in many servers. How would this alternative design compare to the original?

16. FTP servers can be asked to list a directory of files. VSFTPD could do this by calling the system's `ls` (or `dir`) command, displaying the result to a client. But VSFTPD does not do this, and implements directory listings using the relevant system calls directly. Why might you argue that VSFTPD's design makes sense from a security perspective?